

**METHOD AND APPARATUS FOR IMPROVING MESSAGE
AVAILABILITY IN A SUBSYSTEM WHICH SUPPORTS
SHARED MESSAGE QUEUES**

5

CROSS REFERENCES TO RELATED PATENT APPLICATIONS

10

This application claims priority and all other benefits under 35 U.S.C 120 of prior filed co-pending US provisional patent application US Serial No. 60/220,685, filed July 25, 2000, which is incorporated herein by reference.

15

This application is related to the subject matter of the following co-pending patent applications, each of which is assigned to the same assignee as this application and each of which is incorporated herein by reference:

20

D. A. Elko et al., US Serial No. 09/677,339, filed October 2, 2000, entitled "METHOD AND APPARATUS FOR PROCESSING A LIST STRUCTURE" (IBM docket POU920000043);

25

D. A. Elko et al, US Serial No. 09/677,341, filed October 2, 2000, entitled "METHOD AND APPARATUS FOR IMPLEMENTING A SHARED MESSAGE QUEUE USING A LIST STRUCTURE" (IBM Docket POU920000042);

P. Kettley et al., US Serial No. 09/605,589, filed June 28, 2000, entitled "METHOD AND APPARATUS FOR OPERATING A COMPUTER SYSTEM TO ENABLE A RESTART" (IBM docket GB920000031);

5

P. Kettley et al., US Serial No. 60/219,889, filed July 21, 2000, entitled "IMPLEMENTING MQI INDEXED QUEUE SUPPORT USING COUPLING FACILITY LIST STRUCTURES" (IBM docket GB920000033).

10

FIELD OF INVENTION

The present invention relates to methods and apparatus for recovery from failures affecting a resource manager within a data processing network, and has particular applicability to the field of communicating message data between application programs via shared message queues and to the resolution of a failed resource manager's units of work by other active resource managers to improve message availability.

BACKGROUND OF THE INVENTION

In the communication through a computer network of message data between application programs it is known to transmit the messages by means of resource managers such as queue managers which interface to the application programs

25

5

10

through a message queueing interface that is invoked by the application programs. In contemporary data processing environments, it is commonplace for the computer network to connect a client application program that has a task to be performed to one or more transaction-oriented resource manager programs that will undertake the task. In this environment, the client queues an input message through a queuing subsystem to a target system to request processing and when the processing is complete a return message is queued for transmission back to the client.

20

US Patent 5 452 430 describes a data processing system for the storage of persistent and non-persistent data in a queue and a method for the storage of data which is required to survive a system failure. The method involves receiving persistent and non-persistent data to be stored in a queue, then marking the data in time sequence order before storing the persistent data in a first set of data pages and the non-persistent data in a second set of data pages. Upon receiving a request to remove data from the queue, both the first and second pages are checked and the data is removed from the queue in time sequence order. A

log is created to enable recovery in the event of failure and restart of the queue.

It is also known from US Patents 5 797 005 and 5 887 168 to provide a system allowing messages to be processed by any of a plurality of data processing systems in a data processing environment. A shared queue is provided to store incoming messages from message queuing subsystems for processing by one of the plurality of data processing systems. A common queue server receives and queues the messages from a subsystem onto the shared queue so that they can be retrieved by a system having available capacity to process the messages. A system having available capacity retrieves the queued message, performs the necessary processing and places an appropriate response message back on the shared queue. Thus, the shared queue stores messages sent in either direction between clients requesting processing and the data processing systems that perform the processing. Because the messages are enqueued onto the shared queue, the messages can be processed by an application running on any of a plurality of systems having access to the queue. Automatic workload sharing and processing redundancy is provided by this arrangement. If a particular application that is processing a message fails, another application can retrieve that message from the shared queue and perform the processing without the client

having to wait for the original application to be restarted.

SUMMARY OF THE INVENTION

5

It is the aim of the present invention to improve recovery from a connection failure between a resource manager such as a queuing subsystem and a shared resource such as a shared queue, such failure being caused either by communications link failure, or failure of the computer system or of a computer program comprising the resource manager.

10 15 20

In a first aspect of the present invention, there is provided a method for recovering from failures affecting a resource manager within a group of resource managers, wherein the resource managers within the group have access to a shared resource via which remote resource managers communicate with the resource managers within the group, the shared resource including data storage structures to which resource managers within said group connect to send and receive communications, the method comprising:

20

storing, within a first data storage structure of the shared resource, unit of work descriptors for operations

25

performed in relation to said shared resource by the resource managers in said group;

5 sending a notification of a connection failure between a second data storage structure of the shared resource and a first resource manager within said group, the notification being sent to the remaining resource managers within the group which are connected to the second data storage structure;

10 one or more of said remaining resource managers accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

15 said one or more remaining resource managers recovering the identified units of work.

20 In one embodiment, there is provided a method of communicating information relating the state of units of work and the messages which form part of the unit of work between message queuing subsystems coupled together through 25 a coupling facility, the method comprising: communicating the message data in at least one shared queue between the

message queuing subsystems by means of data structures contained in the coupling facility; notifying a connection failure between a queuing subsystem and the coupling facility data structure containing the shared queue, the notification being provided to the remaining queuing subsystems in communication with the coupling facility; including, within the data structure containing the shared queue, information describing which shared queues within the structure are in use from a particular queuing subsystem; including within the said data structures an administrative structure listing unit of work descriptors describing operations performed by the queuing subsystems on those shared queues which are contained within a certain coupling facility data structure; including with the message data stored in the coupling facility data structures, a key to enable retrieval of the message data, one range of key values identifying a message as committed, a further range of key values identifying a message as uncommitted, and a third range of key values identifying the message state as indeterminate; after a connection failure, employing the said remaining queuing subsystems in parallel to interrogate the listed work descriptors so as to identify and to share between them the recovery of units of work active in the subsystem whose connection to the coupling facility has failed, and employing each of the said remaining subsystems to recover its share of the units

of work active in the queuing subsystem whose connection to the coupling facility has failed.

5 After all unit of work descriptors are processed, a second phase of recovery is initiated in which the said remaining subsystems find all inflight messages read and written by the failing queuing subsystem and roll them back. These inflight messages are found without reference to the failed queuing subsystem's logs.

10 In a second aspect of the invention, there is provided a method for recovering from failures affecting a resource manager within a group of resource managers, wherein the resource managers within the group have access to a shared resource, the shared resource including data storage structures to which resource managers within said group connect to perform operations in relation to data held in said shared resource, the method comprising:

20 storing, within a first data storage structure of the shared resource, unit of work descriptors for operations performed by the resource managers in said group in relation to data held in said shared resource;

25 sending a notification of a connection failure between a second data storage structure of the shared resource and

a first resource manager within said group, the notification being sent to the remaining resource managers within the group which are connected to the second data storage structure;

5

one or more of said remaining resource managers accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450

and receive communications to and from remote resource managers, the shared access resource including:

5 means for storing, within a first data storage structure of the shared resource, unit of work descriptors for operations performed in relation to said shared resource by the resource managers in said plurality; and

10 means for sending a notification of a connection failure between a second data storage structure of the shared resource and a first resource manager within said plurality, the notification being sent to the remaining resource managers within the plurality which are connected to the second data storage structure;

wherein said remaining resource managers include:

20 means for accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

25 means for recovering the identified units of work.

In a further aspect of the invention, there is provided a computer program product comprising program code recorded on a machine-readable recording medium, the program code comprising the following set of components:

5

a plurality of resource managers;

10

a shared access resource manager including program code for managing storage and retrieval of data within data storage structures to which the resource managers connect to send and receive communications to and from remote resource managers, the shared access resource manager including:

20

means for storing, within a first data storage structure of the shared resource, unit of work descriptors for operations performed in relation to said shared resource by the resource managers in said plurality; and

25

means for sending a notification of a connection failure between a second data storage structure of the shared resource and a first resource manager within said plurality, the notification being sent to the remaining resource managers within the plurality which are connected to the second data storage structure;

wherein said remaining resource managers include:

5 means for accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

means for recovering the identified units of work.

10

According to another embodiment of the invention, there is provided a communication system to communicate information relating the state of units of work and the messages that form part of the unit of work between message queueing subsystems, the system comprising: a coupling facility to communicate the message data between the message queueing subsystems in at least one shared queue by means of data structures included within the coupling facility; means to notify a connection failure between a queueing subsystem and the coupling facility data structure containing the shared queue, the notification being provided to the remaining queueing subsystems in communication with the coupling facility; the data structure containing the shared queue including information describing which shared queues within the structure are in use from a particular queueing subsystem; an administrative

20

25

structure within the data structures listing unit of work descriptors describing operations performed by the queuing subsystems on those shared queues which are contained within a certain coupling facility data structure; the message data stored in the coupling facility data structures including a key to enable retrieval of the message data, one range of key values identifying a message as committed, a further range of key values identifying a message as uncommitted, and a third range of key values identifying the message data as indeterminate; means operative, after a connection failure, to employ the said remaining queueing subsystems in parallel to interrogate the listed work descriptors so as to identify and to share between them the recovery of units of work active in the subsystem whose connection to the coupling facility has failed and to employ each of the said remaining subsystems to recover its share of the units of work active in the queueing subsystem whose connection to the coupling facility has failed.

10

3

3

0

3

0

4

2

0

2

0

3

5

2

0

3

2

0

3

5

2

0

3

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

0

3

5

2

BRIEF DESCRIPTION OF THE DRAWINGS

5 Embodiments of the invention will now be described in more detail, by way of example, with reference to the accompanying drawings in which;

Figure 1 shows a message queue shared between a plurality of application programs;

10 Figure 2 shows a message queuing system according to the present invention to communicate message data between application programs connected to the system;

15 Figure 3 shows an application structure included within the system of Figure 2;

Figure 4 shows an administrative structure included within the system of Figure 2; and

20 Figure 5 shows steps performed in recovering units of work active in a failed connection to a shared queue.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

25 The present invention is directed to methods and systems to communicate message data between application programs connected to a system and to recover units of

work, consisting of messages read and written, owned by the failing queuing subsystem in parallel by surviving queuing subsystems without access to logs maintained by the failed queuing subsystem.

5

Figure 1 shows a shared queue 10 which receives messages 11, 12, 13 put onto the queue by application programs 14, 15, 16. Programs 17, 18 and 19 get messages from the shared queue 10. The programs 14, 15 and 16 represent client application programs including instructions for performing client-related tasks. When a client program has a transaction to be processed it places a message on the shared queue 10 to be retrieved by one of the programs 17, 18 and 19. When one of the programs 17, 18 and 19 gets a message, it processes the message and places a response on the shared queue 10. Each message remains on the queue until it is retrieved. It will be understood that the diagram of Figure 1 has been simplified for ease of explanation. In practice, a network of connections could be used to communicate messages between the programs requiring to communicate with one another and whilst only one shared queue 10 has been shown for simplicity of explanation, in practice a number of such shared queues are used to pass messages between the communicating programs.

10

11
12
13
14
15
16
17
18
19

20

25

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

Referring now to Figure 2, message queuing subsystems 20 and 21 are provided to interface the programs 14, 15, 16, 17, 18 and 19 to shared queue 10. Whilst only two message queuing subsystems have been shown in Figure 2, it will be apparent to one skilled in the art that the number of subsystems may vary. A coupling facility 22 is provided to couple the message queuing subsystems 20 and 21. A coupling facility manager component 23 is included in each message queuing subsystem to interface to the coupling facility 22 through the use of services provided by the operating system.

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

Each application program 14, 15, 16, 17, 18 and 19 operates on a computer platform comprising a computer and an operating system. The computer typically includes various hardware components, such as one or more central processing units (CPU), a random access memory (RAM) and an input/output (I/O) interface. The message queue subsystems 20 and 21 also run on a computer platform comprising a computer and an operating system and connected to interface to the application programs 14, 15, 16, 17, 18 and 19 either locally or through a communication network which may be implemented as a local area network (LAN) or a wide area network (WAN) for example.

25

The coupling facility 22 runs on a further computer platform including a computer and an operating system. A

5

10

15
16
17
18
19
20
21
22
23
24
25

computer program product comprising instructions stored on a computer readable medium enables the computer to execute the features of the coupling facility as will be discussed below. The coupling facility sets aside an area of storage to hold the messages in the shared queue 10 on their way from one program to another. The communications link between the coupling facility 22 and the connecting computer systems running the message queuing subsystems 20, 21 is managed by a component (referred to as XES) of the operating system of each of the connecting computer systems. The message queuing subsystems 20, 21 and the coupling facility 22 may all run on the same computer or same computer platform.

20

25

The XES operating system component provides functions that allow the message queuing subsystems 20 and 21 to allocate data structures within the coupling facility 22 and to connect to and manipulate these data structures. The XES component also provides status information concerning the status of concurrent connections to the same data structures within the coupling facility 22. The coupling facility managers 23 move data between the coupling facility and the message queuing subsystems 20 and 21. Through the use of the XES services, the coupling facility manager component of the queuing subsystem connects to a single administrative structure 25 used for communicating

unit of work information between the queuing subsystems, and one or more application structures 24 to hold the application message data. Each queuing subsystem sharing message queues will have a connection to the administrative structure and connections to the application structures which hold message data associated with the shared queues being operated on by applications attached via that queuing subsystem. These connections are indicated by the arrows in Figure 2.

10

Referring now to Figure 3, the application structure 24 will be described. The application structure contains an array of list headers 26. Each of the list headers 26 may have anchored to it zero or more list entries that contain message data put by an application interfacing with a queuing subsystem to a shared queue. The list entries each have a key and the list entries are listed under each list header in the order of their keys. The construction of the keys and mechanism for storing and retrieving messages using these is described in commonly assigned US patent application Serial No. 09/677,341, which is incorporated herein by reference (IBM docket POU920000042). USSN 09/677,341 describes how keys represent committed, uncommitted, or 'indeterminate' state messages. The keys also specify the queue manager (queueing subsystem) which put the message on the queue and/or of the queue manager

25

which performed a get operation to retrieve the message, and a timestamp of when the message arrived on the queue. The lowest keyed entry in a list is at the head of its list and the highest keyed entry at the tail of its list. A first set 27 of the list headers is a set of shared queue list headers corresponding respectively to the shared queues 10. A second set 28 of the list headers will be referred to as uncommitted get queue list headers. Each queuing subsystem connected to a coupling facility data structure 22 has its own unique uncommitted get queue. Making entries to the uncommitted get queue lists is explained below.

A list header 29 at the top of the array of list headers has a list of data representing a list header information map 30. For each message queuing subsystem, the information map 30 identifies the shared queue list headers in the array 27 currently being used by that message queuing subsystem. Each queuing subsystem has its own list header information map (LHIM). The LHIM is bound to a specific queuing subsystem based on the LHIM's list entry key. There is a bit string in the LHIM and a one to one correspondence between bits in the bitstring and list headers in the data structure such that, if bit 1 is set on, then the queuing subsystem which owns the LHIM has an interest in the shared queue which maps to list header 1

5

10

15

20

and so on. The coupling facility manager 23 of each message queuing subsystem maintains the entries in the information map 30 for that message queuing subsystem. When a program puts a message onto a shared queue, the coupling facility manager 23 interfacing to that program moves the message data from private storage in the message queuing subsystem to a list entry with a key value in the uncommitted key range in the coupling facility data structure and associates it with the list header in the array 27 identified with that shared queue. When a program gets a message from a shared queue, the coupling facility manager 23 finds the list entry with the lowest key value in the committed range associated with the list header in the array 27 corresponding to the shared queue and copies the message data into private storage in the message queuing subsystem. At the same time, the list entry is dissociated from the list from which it was copied and reassociated with the uncommitted get queue list corresponding to the message queuing subsystem, and the list entry key value is changed to a value in the uncommitted range. Thus messages forming part of an uncommitted unit of work have list entry key values in the uncommitted range.

25

So far, only simple put and get operations have been described. In practice, an application program will perform a series of linked operations to perform a unit of work. A

unit of work is a term of art that refers to a recoverable series of operations performed by an application between two points of consistency. A unit of work begins when a transaction starts or after a user-requested

5 synchronisation point (syncpoint). It ends either at a user-requested syncpoint or at the end of a transaction. A unit of work may involve an application performing a set of operations such as getting a message from a queue, making a database entry based on the contents of the message and putting a different message back onto a queue indicating the results of the database operation. In a cash machine application for example, the cash machine may send a message indicating the withdrawal of a particular amount of cash, the computer at the bank will retrieve the message, debit the appropriate account and send back the current balance. In real life, a program will want either all or none of these steps to occur. So if the database update fails, for example, the original request message should be put back into the queue so that another program may try again.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
20

When the thread of operations that are executed in a unit of work reaches a synchronisation point, they can either be done (known in the art as committed) or undone (known in the art as backed-out or aborted). When an application is part way through the thread of operations in

25

5

a unit of work, it is known as *inflight* so that if it abnormally terminates, the message queuing subsystem can detect this and is able to back-out the updates made by the application. The sequence of operations may then be retried from the beginning by another program.

10

A unit of work may be also be in a fourth state known in the art as '*indoubt*'. The '*indoubt*' state is associated with two phase commit protocols and indicates that the queuing subsystem is unable to make the decision as to whether the unit of work should be committed or backed out as the unit of work (typically) involves other resource managers, such as a database, and must be coordinated by an external syncpoint manager.

15

20

25

The present invention aims to provide sufficient information in the shared storage within the coupling facility 22 that if one of the message queuing subsystems in one computer fails, another message queuing subsystem in another computer is able to back-out or commit the units of work in progress on the failed computer at the time of failure. Figure 4 shows the administrative structure 25 that is used to provide the shared storage for information about operations performed on shared queues in units of work. For units of work progressing beyond the *inflight* state, a unit of work descriptor is built for each coupling

5

10

15

20

25

facility list structure accessed. The unit of work descriptor (UOWD) identifies the list entries containing message data which have been affected by this unit of work in the corresponding coupling facility list structure, and the operation, either a get or put, which has been performed. For example, if the unit of work has accessed multiple coupling facility structures (because for example shared queue 1 maps to structure 1 and shared queue maps to structure 2 and the application did a put to each queue) multiple unit of work descriptors are written - one (or more) for each coupling facility structure accessed by the unit of work. In this case, each unit of work descriptor points to a summary list entry that identifies the set of coupling facility structures the unit of work accessed. For a commit operation, inflight messages put in the unit of work have their list entry key value modified so that it has a value in the committed range, and inflight messages touched by a get operation are deleted from the uncommitted get list header. For a backout operation, inflight messages put in the unit of work are deleted, and inflight messages got in the unit of work are moved back from the uncommitted get queue to the list header from which they had been gotten, and the key value of the list header changed from uncommitted back to a value in the committed range. It is clear that with information about which list entries have been affected in the unit of work, and the

operation (put or get) performed, these operations can be performed by a different queuing subsystem after a failure. As an optimization, no information is kept in the administrative structure about inflight units of work.

5

The administrative structure 25 has an array 31 of list headers. Each of the list headers may have a list of data entries associated with it in the manner already described with reference to the application structure of Figure 3. The array 31 also includes unit of work list headers 34. Each message queuing subsystem is assigned a unit of work list header 34. The UOWD is written into the administrative structure just before the syncpoint operation and associated with the unit of work list header 34 of the queueing subsystem on which the unit of work has been performed. After the syncpoint operation has been performed the UOWD is no longer required and is deleted.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

Referring now to Figure 5, the operations of the message queuing subsystems 20 and 21 and the coupling facility 22 will be described in relation to the recovery of units of work after the failure of a connection between the queuing subsystem and the application structure of Figure 3. In a first step 35, a connection to an application structure in a coupling facility fails. The XES program issues in step 36 a notification of the failure to

any remaining connectors to the same application structure. The remaining connectors may be referred to as peer connectors.

5 Each peer, on receiving a notification from XES that a connection failure has occurred, attempts to start peer recovery by using XES services to broadcast to all peers an instruction to start peer recovery. XES will ensure that only one of these requests is successful and that all peers receive such a request. Peer recovery takes place in two phases, in the first phase units of recovery for the failed connection which are represented by UOWDs associated with the unit of work list header for the queuing subsystem whose connection has failed are recovered. In the second phase, inflight units of work are recovered. Again XES services are used to ensure that phase 1 has completed in all peers before phase 2 can start in any peers.

10 Each peer receives in step 37 a request to start peer recovery for the failed connector for the application structure. The process enters a phase 1 of recovery in which the unit of work list header for the failed connector is established in step 38 from information contained in the connection identifier for the failed connection which is notified to the surviving connectors by XES. In step 39 the peers work in parallel through the units of work

15
20
25
30
35
40
45
50
55
60
65
70
75
80
85

25

descriptors in the list for the failed connector and each selects a unit of work for recovery (when a peer finds there are no further units of work to recover, it has completed phase 1). The ownership and recovery of an individual unit of work is effected through the version number mechanism described in the attached design documentation. In step 40 each peer recovers the unit of work that it has selected in its entirety on its own and then makes a selection of the next unit of work that is available for recovery (i.e. not selected and owned by another peer). The recovery of a unit of work involves completing the commit for units of work marked in-commit, backing out units of work marked in-abort, or moving to an 'indeterminate' key range, the messages in a unit of work marked as in-doubt. The peers thus co-operate in parallel together, selecting a single unit of work each at a time for recovery and working down the list of unit of work descriptors in the list. When there are no further unit of work descriptors associated with the queuing subsystem whose connection has failed left to process, a peer indicates that it has finished phase 1 of recovery. In step 41 each peer issues a confirmation to the XES program that it has completed phase 1 of recovery and in step 42 the XES program issues to all the peers a notification when all the peers have confirmed the completion of phase 1 of recovery so that the peers can all start phase 2.

5

10

15

20

25

5

10

15

20

25

In a second phase of recovery, the recovery work to be performed is BACK-OUT of inflight activity. In step 43, each peer refers to the information map 30 in the application structure to find the shared queue list headers that may have inflight activity. In step 44, each peer selects a list header for inflight recovery and prevents other peers from also using it through the use of an ENQ (serialization mechanism). Inflight recovery involves deleting list entries with a key in the uncommitted keyrange which were put by the queuing subsystem whose connection to the coupling facility has failed, and finding list entries with a key in the uncommitted key range on the uncommitted get queue and which were originally associated with the shared queue being processed and moving them back to that list header, at the same time, changing the key value of the list entry to be in the committed key range. In step 45, each peer recovers the entries under the selected list header entirely on its own and then deals with the next available list header. The peers co-operate together in parallel to each recover a list header at a time. When a peer has recovered a list header, it updates the entry in the information map 30 to reset the interest of the failed connection in that list header. At the end of phase 2, the information map 30 shows that the failed connection has no interest in any of the list headers. No

further recovery work can be performed by peers, although on restart of the failed queueing subsystem there may be indoubt messages whose list entries had been moved to keys in the 'indeterminate' key range by peer recovery and which can be either committed or backed out after reference to the restarting queueing subsystem's log or the external syncpoint manager. Each peer notifies XES that it has completed phase 2.

Because of the parallel nature of the recovery performed by the peer connectors, it is possible to undertake the recovery of more than one failed connection to a shared queue. Thus if a connection A fails and peer connections B and C survive, the peer connections B and C operate in parallel to recover the connection A. If peer B should fail during the recovery of A, the remaining peer C will complete the recovery of A and then proceed to recover peer B, including any recovery which B was doing for A as indicated by B's ownership of UOWDs on A's unit of work list header.

What has been described above is the processing performed by surviving queueing subsystems with respect to a single application structure. In general, the failed queueing subsystem may have been connected to multiple application structures, for example structure 1 and 2. If a

5

surviving queuing subsystem is also connected to both structures 1 and 2, it receives two notifications from XES, one for structure 1 and another for structure 2. The surviving queuing subsystem in this case is able to perform the recovery processing (as described above) in parallel for the shared queues that map to structure 1 and structure 2.

10

This results in the ability to recover a multiplicity of shared queues in parallel, the set of shared queues distributed over a set of application structures provided a surviving peer is connected to the same set (or a subset) of the structures to which the failing queuing subsystem had a connection.

15

20

25

What has been described is a method and system for communicating unit of work information between a plurality of message queuing subsystems. The unit of work state information being provided either through the use of keys, or explicitly through Unit of Work Descriptors stored in the administrative structure. This communication of unit of work information means that in the event of failure of a message queuing subsystem or its communication with the shared queues, other message queuing subsystems can progress the active units of work without reference to the failing subsystem's log data or waiting for the failed

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

135

140

145

150

155

160

165

170

175

180

185

190

195

200

205

210

215

220

225

230

235

240

245

250

255

260

265

270

275

280

285

290

295

300

305

310

315

320

325

330

335

340

345

350

355

360

365

370

375

380

385

390

395

400

405

410

415

420

425

430

435

440

445

450

455

460

465

470

475

480

485

490

495

500

505

510

515

520

525

530

535

540

545

550

555

560

565

570

575

580

585

590

595

600

605

610

615

620

625

630

635

640

645

650

655

660

665

670

675

680

685

690

695

700

705

710

715

720

725

730

735

740

745

750

755

760

765

770

775

780

785

790

795

800

805

810

815

820

825

830

835

840

845

850

855

860

865

870

875

880

885

890

895

900

905

910

915

920

925

930

935

940

945

950

955

960

965

970

975

980

985

990

995

1000

queueing subsystem to restart, so improving message availability. The maintenance of the list header information map means that the queueing subsystem performing the recovery can rapidly find any inflight units of work in progress at the time of failure, and back them out without the overhead of explicitly maintaining in a shared place the list of messages in the unit of work. A plurality of message queueing subsystems 20, 21 interface to the application programs and are coupled together through one or more coupling facilities 22. The functioning of the subsystems 20 and 21 and the coupling facility 22 are illustrative of the functions to be performed in communicating message data between the application programs to which the subsystems interface. The message data is communicated in shared queues between the message queueing subsystems by means of the data structures 24 contained in the coupling facility. The application structure 24 lists shared queues to which each message queueing subsystem is connected and the administrative structure 25 lists unit of work descriptors describing operations performed by the queueing subsystems on the shared queues for units of work which have reached a syncpoint operation. A connection failure between a queueing subsystem and a shared queue is provided to the remaining queueing subsystems connected to the shared queue. The remaining queueing subsystems operate in parallel to interrogate the listed work descriptors so

as to identify and to share between them the units of work active in the failed connection, and each of the said remaining subsystems recovers its share of the units of work active in the failed connection.

5

The solution described above includes using a structure interest map in a second phase of recovery, to identify the Coupling Facility list structures to which a resource manager (queuing subsystem) was connected. The second phase of recovery is carried out when a resource manager of the group is connected to the relevant list structures. In an alternative embodiment, a more proactive recovery scheme is used which involves one or more resource managers within the group referring to the structure interest map to determine whether additional connections should be established. In this embodiment, if a failed resource manager performed uncommitted operations in relation to a list structure to which no other resource managers were connected, other resource managers within the group will establish a connection and perform the required recovery.

10
15
20